

μ Analysis with Real Parametric Uncertainty

Peter M. Young Matthew P. Newlin John C. Doyle

Electrical Engineering, 116-81
California Institute of Technology
Pasadena, CA 91125, U.S.A.

Abstract

This paper gives a broad overview, from a LFT/ μ perspective, of some of the theoretical and practical issues associated with robustness in the presence of real parametric uncertainty, with a focus on computation. Recent results on the properties of μ in the mixed case are reviewed, including issues of NP completeness, continuity, computation of bounds, the equivalence of μ and its bounds, and some direct comparisons with "Kharitonov-type" analysis methods. In addition, some advances in the computational aspects of the problem, including a new Branch and Bound algorithm, are briefly presented together with numerical results. The results of this paper strongly suggest that while the mixed μ problem may have inherently combinatoric worst-case behavior, practical algorithms with modest computational requirements can be developed for problems of medium size (< 100 parameters) that are of engineering interest.

1 Introduction

The analysis of systems subject to mixed real (eg. parametric) and complex (eg. dynamic) structured uncertainty has been the focus of much research interest in recent years. Many robust stability and performance problems associated with such systems can be addressed with the structured singular value, μ , by choice of an appropriate (mixed) block structure. This mixed μ problem can have fundamentally different properties from the more standard complex μ problem (where the block structure contains only complex blocks), and these properties have important implications for computation.

In section 3 we review the results in [13, 1, 2], which show that both the real and mixed μ problems are, in general, both discontinuous in the problem data and NP complete. While these results have devastating implications for certain research directions, they need not preclude the development of practical algorithms for medium size problems. Here medium size means less than 100 real parameters, and "practical" means avoiding combinatoric (nonpolynomial) growth in computation with the number of parameters for all of the problems which arise in engineering applications.

The rest of the paper is devoted to demonstrating that such practical algorithms are not only possible, but appear to be available, at least in experimental form. Section 4 reviews the bounds for mixed μ introduced in [4, 3], which play a key role in the proposed algorithm. Several important properties of the bounds are described in Section 5. One of the more interesting properties is related to recent work ([15]) which shows that several celebrated "Kharitonov-type" results are subsumed by the methods emphasized in this paper, and can be computed almost trivially without any need to examine a combinatoric number of vertices or edges. It also sheds light on why there have been no practical algorithms based on "Kharitonov-type" theorems.

Section 6 describes some practical schemes to compute the bounds described in section 4 and explores the algorithm's performance in trials on large collections of data. Both the quality of the bounds and their computational requirements as a function of problem size are explored. While the bounds are often accurate enough for engineering purposes, in a significant number of cases of interest, they are not. This is in contrast with the purely complex nonrepeated case, where no examples of problems with large gaps have been found.

The use of Branch and Bound schemes to improve upon existing

bounds has been suggested by several authors (see [5, 6] and references therein), and section 7 considers the use of such schemes. There are some important issues and tradeoffs to be considered in implementing such a scheme, which can greatly impact the performance. A selection of results from a fairly extensive numerical study of these issues is presented, and a Branch and Bound scheme is proposed which we believe will form the basis of a practical scheme. The material in sections 5 through 7 will be covered in greater depth in future papers [7, 8, 9].

2 Notation and Definitions

The notation used here is fairly standard and is essentially taken from [10] and [3]. For any square complex matrix M we denote the complex conjugate transpose by M^* . The largest singular value and the structured singular value are denoted by $\bar{\sigma}(M)$ and $\mu_K(M)$ respectively. The spectral radius is denoted $\rho(M)$ and $\rho_R(M) = \max\{|\lambda| : \lambda \text{ is a real eigenvalue of } M\}$, with $\rho_R(M) = 0$ if M has no real eigenvalues. For a Hermitian matrix M , then $\bar{\lambda}(M)$ and $\lambda_{\min}(M)$ denote the largest and smallest (real) eigenvalues respectively. For any complex vector x , then x^* denotes the complex conjugate transpose and $\|x\|$ the Euclidean norm. We denote the $k \times k$ identity matrix and zero matrix by I_k and O_k respectively.

The definition of μ is dependent upon the underlying block structure of the uncertainties, which is defined as follows. Given a matrix $M \in \mathbb{C}^{n \times n}$ and three non-negative integers m_r , m_c , and m_C with $m := m_r + m_c + m_C \leq n$, the block structure $K(m_r, m_c, m_C)$ is an m -tuple of positive integers

$$K = (k_1, \dots, k_{m_r}, k_{m_r+1}, \dots, k_{m_r+m_c}, k_{m_r+m_c+1}, \dots, k_m) \quad (1)$$

where we require $\sum_{i=1}^m k_i = n$ in order that the dimensions are compatible. This determines the set of allowable perturbations, namely define

$$X_K = \{\Delta = \text{block diag}(\delta_1^r I_{k_1}, \dots, \delta_{m_r}^r I_{k_{m_r}}, \delta_1^c I_{k_{m_r+1}}, \dots, \delta_{m_c}^c I_{k_{m_r+m_c}}, \Delta_1^C, \dots, \Delta_{m_C}^C) : \delta_i^r \in \mathbb{R}, \delta_i^c \in \mathbb{C}, \Delta_i^C \in \mathbb{C}^{k_{m_r+m_c+i} \times k_{m_r+m_c+i}}\} \quad (2)$$

Note that $X_K \in \mathbb{C}^{n \times n}$ and that this block structure is sufficiently general to allow for repeated real scalars, repeated complex scalars, and full complex blocks. Note also that the full complex blocks need not be square, but we restrict them as such for notational convenience. The purely complex case corresponds to $m_r = 0$.

Definition 1 ([11]) The structured singular value, $\mu_K(M)$, of a matrix $M \in \mathbb{C}^{n \times n}$ with respect to a block structure $K(m_r, m_c, m_C)$ is defined as

$$\mu_K(M) = \left(\min_{\Delta \in X_K} \{\bar{\sigma}(\Delta) : \det(I - \Delta M) = 0\} \right)^{-1} \quad (3)$$

with $\mu_K(M) = 0$ if no $\Delta \in X_K$ solves $\det(I - \Delta M) = 0$.

In order to develop the upper and lower bound theory for μ we need to define some sets of block diagonal scaling matrices (which are also dependent on the underlying block structure).

$$Q_K = \{\Delta \in X_K : \delta_i^r \in [-1, 1], \delta_i^c \delta_i^{c*} = 1, \Delta_i^{C*} \Delta_i^C = I_{k_{m_r+m_c+i}}\} \quad (4)$$

$$D_K = \{\text{block diag}(D_1, \dots, D_{m_r+m_c}, d_1 I_{k_{m_r+m_c+1}}, \dots, d_{m_C} I_{k_m}) : 0 < D_i = D_i^* \in \mathbb{C}^{k_i \times k_i}, 0 < d_i \in \mathbb{R}\} \quad (5)$$

$$\mathcal{G}_K = \{\text{block diag}(G_1, \dots, G_{m_r}, O_{k_{m_r+1}}, \dots, O_{k_m}) : \\ G_i = G_i^* \in \mathbb{C}^{k_i \times k_i}\} \quad (6)$$

$$\hat{\mathcal{D}}_K = \{\text{block diag}(D_1, \dots, D_{m_r+m_c}, d_1 I_{k_{m_r+m_c+1}}, \dots, d_{m_c} I_{k_m}) : \\ \det(D_i) \neq 0, D_i \in \mathbb{C}^{k_i \times k_i}, d_i \neq 0, d_i \in \mathbb{C}\} \quad (7)$$

$$\hat{\mathcal{G}}_K = \{\text{block diag}(g_1, \dots, g_{n_r}, O_{n_c}) : g_i \in \mathbb{R}\} \quad (8)$$

where $n_r = \sum_{i=1}^{m_r} k_i$ and $n_c = n - n_r$.

3 Properties of the Mixed μ Problem

The mixed μ problem inherits many of the properties of the complex μ problem, such as the scaling property and invariance to appropriate block diagonal similarity transformations (see [11, 3]). However in some aspects the mixed μ problem can be fundamentally different from the complex μ problem, and in this regard we consider two issues in the remainder of this section: NP completeness and continuity. This material is essentially taken from [1, 2].

3.1 Continuity

It is now well known that real μ problems can be discontinuous in the problem data (see [12]). As well as adding computational difficulties to the problem this sheds serious doubt on the usefulness of real μ as a robustness measure in such cases, since the system model is always a mathematical abstraction from the real world, and is computed to finite precision. It is shown in [2] how to regularize these problems by essentially adding a small amount of complex uncertainty to each real uncertainty. By doing this a small amount of phase uncertainty is added to the gain uncertainty. It is then shown that the new mixed μ problem is continuous. This regularization seems reasonably well motivated from an engineering point of view, where unmodeled dynamics would always produce some phase uncertainty.

Furthermore it is shown in [2] that mixed μ problems containing some complex uncertainty are, under some mild assumptions, continuous even without the regularization procedure outlined above (whereas purely real μ problems are not). This is reassuring from an engineering viewpoint since one is usually interested in robust performance problems (which therefore contain at least one complex block), or robust stability problems with some unmodeled dynamics, which are naturally covered with complex uncertainty. Thus in problems of engineering interest, the potential discontinuity of μ should not arise, although conditioning of μ computation could be a problem and needs more study.

3.2 NP Completeness

Recent results in [13] show that a special case of computing μ with real perturbations only is NP complete. While these results do not apply to the complex only case, it is certainly true that the general mixed problem is NP complete as well. Demmel's results [1] also cast doubt on the tractability of ϵ -approximations to the pure real case, although the mixed case is less clear. These results strongly suggest that it is futile to pursue exact methods for computing μ in the purely real or mixed case for even moderate (less than 100) numbers of real perturbations, unless one is prepared not only to solve the real μ problem but also to make fundamental contributions to the theory of computational complexity. Furthermore, it may be that even approximate methods must have worst-case combinatoric complexity.

These results do not mean, however, that practical algorithms are not possible. Practical algorithms for other NP complete problems exist and typically involve approximation, heuristics, branch-and-bound, or local search. The results of this paper strongly suggest that an intelligent combination of all these techniques can yield a practical algorithm for the mixed problem. On the other hand, routine application of any of these methods generally seems to produce algorithms with clearly exponential growth rates even on small problems. For example, using Branch and Bound with crude bounds ([5]) appears to have exponential growth on typical problems.

4 Upper and Lower Bounds for Mixed μ

Whilst definition 1 gives an exact expression for μ it does not yield directly any tractable way to compute μ , since the associated optimization problem is not in general convex. For this reason we are interested in computing upper and lower bounds for mixed μ problems.

First consider the computation of a lower bound. Note that one cannot simply 'cover' the real perturbations with complex ones (and then use the complex μ lower bound) since that would include perturbations from outside the permissible set X_K , and so would not yield a valid lower bound. The key to obtaining a lower bound lies in the fact that the μ problem may be reformulated as a real eigenvalue maximization. The following theorem is taken from [3].

Theorem 1 ([3]) For any matrix $M \in \mathbb{C}^{n \times n}$, and any compatible block structure K

$$\max_{Q \in \mathcal{Q}_K} \rho_R(QM) = \mu_K(M) \quad (9)$$

This immediately gives us a theoretical lower bound since we have that for any $Q \in \mathcal{Q}_K$, $\rho_R(QM) \leq \mu_K(M)$. The idea then is to find an efficient way to compute a local maximum of the function $\rho_R(QM)$ over $Q \in \mathcal{Q}_K$. Note that since this function is non-convex we cannot guarantee to find the global maximum and hence we only obtain a lower bound for μ . The practical computation of such a local maximum is discussed in section 6.1.

Now consider an upper bound for μ . One could, for the purposes of the upper bound, cover the real perturbations with complex ones (and then use the complex μ upper bound) since this would cover the admissible perturbation set X_K . However this approach does not exploit the phase information that is present in the real perturbations, and hence the bound is frequently poor. The upper bound presented in [4] does exploit this phase information and gives a bound which is never worse than the standard upper bound from complex μ theory (see [14] for example) and is frequently much better. The following theorem is taken from [4].

Theorem 2 ([4]) For any matrix $M \in \mathbb{C}^{n \times n}$, and any compatible block structure K suppose α_* is the result of the minimization problem

$$\alpha_* = \inf_{\substack{D \in \mathcal{D}_K \\ G \in \mathcal{G}_K}} \left[\min_{\alpha \in \mathbb{R}} \{ \alpha : (M^*DM + j(GM - M^*G) - \alpha D) \leq 0 \} \right] \quad (10)$$

then if $\alpha_* \leq 0$ we have $\mu_K(M) = 0$, otherwise

$$\mu_K(M) \leq \sqrt{\alpha_*} \quad (11)$$

Since the above minimization involves a LMI (Linear Matrix Inequality), it is convex (so that all local minima are global) and hence this bound is computationally tractable. The practical computation of the upper bound is discussed briefly in section 6.2.

5 Properties of the Bounds

The upper and lower bounds from complex μ theory not only serve as computational schemes, but are theoretically rich as well. Connections between the bounds and various aspects of linear system theory have already been established, and further work in this area appears to have great promise. A theoretical study of the mixed μ bounds may yield new insight as well, and this is the subject of current research. One of the problems currently under investigation is to establish for which classes of matrices and block structures the upper bound is identically equal to μ . This is of interest since the upper bound is a convex optimization problem, and hence can be computed exactly. Note that although the lower bound from (9) is *always* equal to μ if one finds the global maximum, it is a non-convex problem (and hence one cannot guarantee to find the global maximum). Some preliminary results in this area are briefly presented in the remainder of this section. This work will be presented in more detail in [7].

5.1 A Theoretical Framework for the Problem

To examine the equivalence between μ and its upper bound, we first consider under what conditions a given pair of scaling matrices, D_0, G_0 , yields a value of $\beta = \sqrt{\alpha}$ which equals the minimum of the upper bound minimization, and under what conditions β equals μ . In the remainder of this subsection we present some of the ideas, and state without proof two theorems which answer these questions.

Suppose we have matrices $M \in \mathbb{C}^{n \times n}$, $D_0 \in \mathcal{D}_K$ and $G_0 \in \mathcal{G}_K$ and a real scalar $\beta > 0$ such that $\bar{\lambda}(M^* D_0 M + j(G_0 M - M^* G_0) - \beta^2 D_0) = 0$ with r eigenvalues coalesced at the maximum. Further suppose that the eigenvectors are given by $U_0 \in \mathbb{C}^{n \times r}$ where $(M^* D_0 M + j(G_0 M - M^* G_0) - \beta^2 D_0) U_0 = 0$ and $U_0^* D_0 U_0 = I_r$. Then the question of whether or not we can find a pair of scaling matrices $D \in \mathcal{D}_K, G \in \mathcal{G}_K$ to improve upon D_0, G_0 can be related to whether or not there exists a vector $\hat{\eta} \in \mathbb{C}^r$, $|\hat{\eta}| = 1$ such that the quantity $\phi_{D,G}(\hat{\eta}) \doteq \hat{\eta}^* U_0^* (M^* D M + j(G M - M^* G) - \beta^2 D) U_0 \hat{\eta}$ satisfies $\phi_{D,G}(\hat{\eta}) \leq 0$. The function $\phi_{D,G}(\eta)$ can in turn be written as $\phi_{D,G}(\eta) = \langle (D, G), (D(\eta), G(\eta)) \rangle$, an inner product between the pair (D, G) and a pair $(D(\eta), G(\eta))$ parametrized by the vector η . This can be used to define a set ∇_Y as the set of all such pairs $(D(\eta), G(\eta))$ (for $|\eta| = 1$), together with an extended set $\hat{\nabla}_Y$ (with $\nabla_Y \subset \hat{\nabla}_Y$). The details of these constructions will be given in [7], and the analogous constructions for the complex μ case are given in [14]. The relationship between the minima of the upper bound function and μ is intimately related to the nature of these two sets, and this is stated explicitly in the following two theorems.

Theorem 3 Suppose we have matrices $M \in \mathbb{C}^{n \times n}$, $D_0 \in \mathcal{D}_K$ and $G_0 \in \mathcal{G}_K$ and a real scalar $\beta > 0$ such that $\bar{\lambda}(M^* D_0 M + j(G_0 M - M^* G_0) - \beta^2 D_0) = 0$ with r eigenvalues coalesced at the maximum. Further suppose that the eigenvectors are given by $U_0 \in \mathbb{C}^{n \times r}$ where $(M^* D_0 M + j(G_0 M - M^* G_0) - \beta^2 D_0) U_0 = 0$ and $U_0^* D_0 U_0 = I_r$. Then D_0, G_0 are minimizing arguments of the upper bound problem (with minimum value $\alpha = \beta^2$)

$$\inf_{D \in \mathcal{D}_K, G \in \mathcal{G}_K} \left[\min_{\alpha \in \mathbb{R}} \{ \alpha : (M^* D M + j(G M - M^* G) - \alpha D) \leq 0 \} \right] \quad (12)$$

if and only if $0 \in \text{Co}(\nabla_Y)$ (where $\text{Co}(\nabla_Y)$ denotes the convex hull of ∇_Y).

Theorem 4 Suppose we have $M \in \mathbb{C}^{n \times n}$, $U_0 \in \mathbb{C}^{n \times r}$ and $\beta > 0$ defined as in theorem 3. Then $\beta = \mu_K(M)$ if and only if $0 \in \hat{\nabla}_Y$.

This type of theoretical framework has been very successful in analyzing the complex μ problem, and it is hoped that further work in this area will establish when which mixed μ equals its upper bound. This is a subject of current research. The following related result was obtained in [4].

Theorem 5 ([4]) Suppose we have $M \in \mathbb{C}^{n \times n}$, then provided the infimum in (10) is achieved and the corresponding largest eigenvalue of $(M^* D M + j(G M - M^* G) - \alpha D)$ is distinct, then $\mu_K(M)$ equals its upper bound from theorem 2.

5.2 The Rank One Case and “Kharitonov-type” results

The mixed μ problem when M is rank one is studied in detail in [15]. The authors develop an analytic expression for the solution to this problem, reducing it to a convex minimization problem in one variable. Not only is this easy to compute, but it has sublinear growth in the problem size. Interestingly, they show that this rank one case corresponds to the so called “affine parameter variation” problem (for a polynomial with perturbed coefficients) which has also been examined in detail in the literature, and for which several celebrated “Kharitonov-type” results have been proven (see [16] for example). It is noted in [15] that all these problems can be treated as special cases of “rank one μ problems” and are thus “relatively easy to solve”. Even the need to check (a combinatoric number of) edges is shown to be unnecessary.

This rank one case can also be addressed within the framework developed here for examining the equivalence between μ and its upper

bound. The following theorem, stated without proof, gives a partial answer to the rank one mixed μ problem.

Theorem 6 Suppose we have a rank one matrix $M \in \mathbb{C}^{n \times n}$, then provided the infimum in (10) is achieved, $\mu_K(M)$ equals its upper bound from theorem 2.

We believe that theorem 6 extends to the general case (where the infimum may not be achieved) and this is being investigated. If this extension holds, then it says that for such problems μ equals its upper bound and is hence equivalent to a convex problem. This reinforces the results of [15] and offers some insight into why the problem becomes so much more difficult when we move away from the “affine parameter variation” case to the “multilinear” or “polynomial” cases [6]. These correspond to μ problems where M is not necessarily rank one, and hence may no longer be equal to the upper bound and so may no longer be equivalent to a convex problem (note that there exist rank two matrices for which μ does not equal its upper bound).

These results also underline why there are no practical algorithms based on “edge-type” theorems, as the results appear to be relevant only to a very special problem. Furthermore, even in the very special “affine parameter case” there are a combinatoric number of edges to check.

5.3 The Mixed μ Upper Bound Versus the Complex μ Upper Bound

As was mentioned in section 4 it is always possible to obtain an upper bound for a mixed μ problem simply by treating the real parameters as complex, and using the standard complex μ upper bound (see [14] for example). However the upper bound from theorem 2 is frequently much better than the complex μ upper bound because of the extra degrees of freedom we have in choosing the G scaling matrix (note that if we restrict ourselves to $G = 0$, we recover the complex μ upper bound). The G scaling matrix exploits the phase information we have about the real parameters in order to reduce the bound. However it is not always possible to improve upon the complex μ upper bound via the G scaling matrix as is illustrated in the following theorem, stated without proof.

Theorem 7 Suppose we have a real matrix $M \in \mathbb{R}^{n \times n}$ and a block structure K with $k_i = 1$ for $i = 1, \dots, m_r$ (i.e. none of the real scalars are repeated) then an optimal choice for G in (10) is $G = 0_n$.

This is an important class of problems. For instance one encounters μ problems where M is real when it is constructed from State Space ‘ A, B, C, D ’ matrices. Note that this theorem does not apply if any of the real parameters are repeated.

6 Practical Computation of the Bounds

The theoretical bounds described in section 4 form the basis of our computation scheme. However a certain amount of reformulation is required before they can be implemented in an efficient manner. This is described briefly in the remainder of this section, together with some numerical experience with the algorithm, and will be presented in greater detail in [8]. The algorithm has been implemented as a Matlab function (m-file), and will be available shortly in a test version in conjunction with the μ -Tools toolbox. While these algorithms are far from optimal, they serve to demonstrate the practicality of this approach, and should thus motivate more refined algorithms.

6.1 The Lower Bound

In order to compute a lower bound for μ we need to find a local maximum of the maximization problem (9) as discussed in section 4. It turns out that this can be done efficiently by means of a power iteration. The iteration scheme usually converges fairly rapidly and each iteration of the scheme is very cheap, requiring only such operations as matrix-vector multiplications and vector inner products. The scheme tested here is a very simple power iteration, and as such does not converge on all problems, but in such cases one still obtains a candidate mixed perturbation from the iteration scheme. From this one can

compute a lower bound (provided that the mixed μ problem contains some complex uncertainty) by simply wrapping in the real perturbations, and then evaluating the spectral radius of the associated complex μ problem, scaled by the candidate complex perturbations. There are several refinements that should improve the convergence and these will be investigated. The theoretical development of the power iteration, together with some aspects of its implementation, is fully described in [3] and we will not go into any of the details here.

6.2 The Upper Bound

Since the upper bound from theorem 2 is convex one could tackle it using a variety of convex programming techniques. For instance we know that gradient search methods will lead us to the minimum eventually, although they may be slow (although the upper bound problem (10) is not in general differentiable if the maximum eigenvalue is repeated, it is possible to compute a generalized gradient which gives a descent direction). Since the upper bound can be formulated as an LMI, it is hoped that specialized algorithms for solving LMIs will emerge that greatly improve the bounds computation. In the meantime, we would like to exploit the specific structure of the problem in a somewhat ad hoc manner in order to speed up the computation. In particular we can reformulate the problem via the following theorem, stated without proof.

Theorem 8 Suppose we have a matrix $M \in \mathbb{C}^{n \times n}$ and a real scalar $\beta > 0$, then there exist matrices $D \in \mathcal{D}_K, G \in \mathcal{G}_K$ such that

$$\bar{\lambda}(M^*DM + j(GM - M^*G) - \beta^2 D) \leq 0 \quad (13)$$

if and only if there exist matrices $\hat{D} \in \hat{\mathcal{D}}_K, \hat{G} \in \hat{\mathcal{G}}_K$ such that

$$\bar{\sigma} \left((I + \hat{G}^2)^{-\frac{1}{2}} \left(\frac{\hat{D}M\hat{D}^{-1}}{\beta} - j\hat{G} \right) (I + \hat{G}^2)^{-\frac{1}{2}} \right) \leq 1 \quad (14)$$

It is clear from this that as an alternative to carrying out the minimization in (10) we could compute the ‘minimum’ $\beta > 0$ such that

$$\inf_{D \in \mathcal{D}_K, G \in \mathcal{G}_K} \bar{\sigma} \left((I + \hat{G}^2)^{-\frac{1}{2}} \left(\frac{\hat{D}M\hat{D}^{-1}}{\beta} - j\hat{G} \right) (I + \hat{G}^2)^{-\frac{1}{2}} \right) \leq 1 \quad (15)$$

Note that the theoretical equivalence of the two problems breaks down at $\beta = 0$ (and so for these cases strictly speaking there is no minimum β) but this presents no problem for a practical computation scheme since we merely quit if the upper bound falls below some prespecified tolerance (which can be arbitrarily small). Each of these two different formulations of the upper bound problem has its own advantages. The problem statement from (13) has the advantages that it is linear in the matrices D and G , and is convex (and hence one will not have problems associated with local minima). The problem statement from (14) has the advantages that one is trying to minimize the norm of a given matrix (which offers some numerical advantages), that D enters the problem exactly as in the standard complex μ upper bound, that G enters the problem in a balanced symmetric fashion, and that G is now a real diagonal matrix.

The upper bound algorithm implemented here works by initially tackling the problem in the form of (14). Here we can use some methods from the complex μ bounds, together with various other techniques, to obtain a fairly good estimates of \hat{D}, \hat{G} and β . These are then converted into an initial guess for the problem in the form of (13) and the algorithm then proceeds to improve on these. This will be covered in greater detail in [8].

6.3 Algorithm Performance

There are many questions one could ask with regard to the algorithm performance, both in terms of computation time and accuracy of the resulting bounds. We decided to focus on the algorithm performance versus matrix size for a fixed set of uncertainty descriptions. The first test performed was to examine the average computation time for the algorithm implemented in Matlab. For this purpose we used random complex matrices generated in Matlab with the μ -Tools “crand” command (having set “rand(‘normal’)”). This generates matrices whose

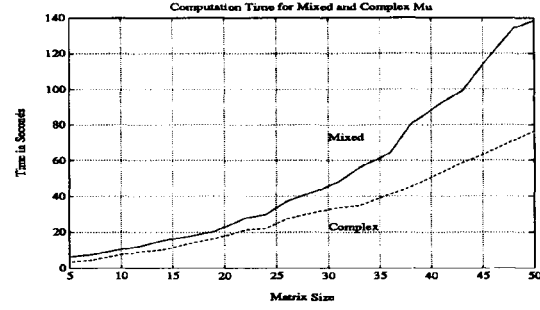


Figure 1: Computation time in seconds versus matrix size for mixed- μ problem (solid) and complex- μ problem (dashed).

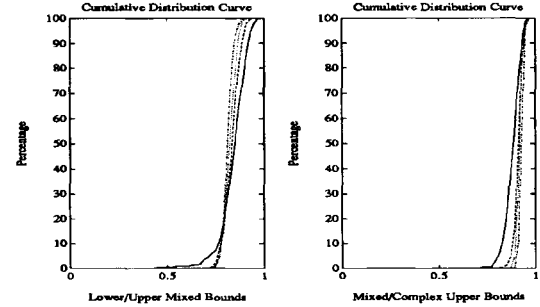


Figure 2: Ratios of mixed- μ lower to upper bounds, and mixed- μ to complex- μ upper bounds, for a sample of random matrices. Matrices of sizes 10 (solid), 20 (dashed), 30 (dotted), and 50 (dashdot).

elements are normally distributed (zero mean) random variables. The computation time is shown versus matrix size in Figure 1 for block structures consisting of all scalar uncertainties, with 80% of them chosen as real and the rest complex. The time to compute the appropriate complex μ problem using the μ -Tools “mu” command is shown for comparison. The results were obtained running Matlab on a Sparc 1 workstation, and it can be seen that we can reasonably expect to handle problems of size 10 in about 10 seconds, up to problems of size 50 in about 2-3 minutes.

The next set of tests performed was aimed at evaluating the accuracy of the bounds. Again we used random complex matrices generated as before, and the same class of block structures. This time we compared the upper and lower mixed μ bounds, and also the mixed μ and complex μ upper bounds. The complex μ bounds were obtained by simply replacing all the real perturbations with complex ones, but without changing the matrix. Thus the complex upper bound is strictly larger than the mixed upper bound. The results are shown in Figure 2, and indicate that for these problems we are obtaining fairly tight bounds, even for large problems.

It is also apparent that for these problems there is typically not much of a gap between mixed μ and complex μ . This class of matrices is interesting from the point of view of the lower bound performance, since one is obtaining a mixed perturbation achieving a lower bound close to that one could obtain with a complex perturbation. However it is not too interesting from the point of view of the upper bound performance, since the G scaling matrix cannot greatly reduce the upper bound. It is also doubtful that random complex matrices are representative of those of practical interest. For these reasons we would like to find a class of matrices where we often encounter problems with a reasonably large gap between mixed μ and complex μ , as well as matrices which are of more practical relevance.

Since the matrices that the μ software will be run on are typically obtained from control problems, a fairly natural class of matrices is

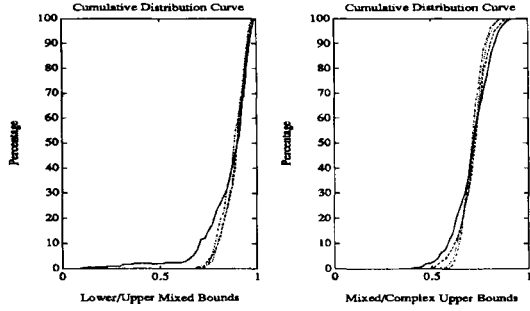


Figure 3: Ratios of mixed- μ lower to upper bounds, and mixed- μ to complex- μ upper bounds, for a sample of random (system) matrices. Matrices of sizes 10 (solid), 20 (dashed), 30 (dotted), and 50 (dashdot).

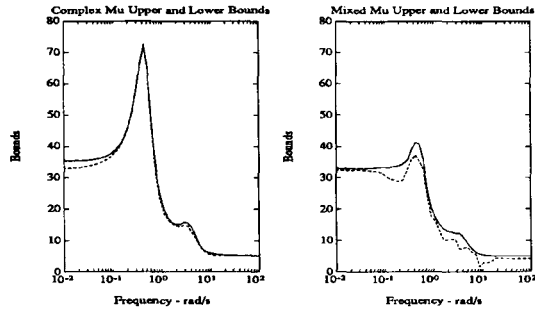


Figure 4: Complex- μ and mixed- μ upper and lower bounds versus frequency for a random system.

to randomly generate State Space 'A,B,C,D' matrices as above, and then evaluate the transfer matrix at some frequency (usually placed roughly in the middle of the modes). The same set of tests as outlined above was performed on this class of matrices, except that the block structures consisted of all scalar uncertainties, with 90% of them chosen as real and the rest complex. The results are shown in Figure 3, and it can be seen that the bounds are once again reasonably tight, but now we are obtaining a wider spread of values for the gap between complex μ and mixed μ , providing a better test of the upper bound performance.

As a final test the bounds for the mixed μ problem were evaluated across a frequency range for systems generated as above, and compared to the bounds for the appropriate complex μ problem. Again the bounds seemed reasonably tight, and a typical example plot is shown in Figure 4.

Note that all these tests were aimed at evaluating the typical performance of the algorithm on an essentially random selection of problems, and it appears that the algorithm is performing well for most problems. This does not mean however that one cannot encounter mixed μ problems where the gap between the upper and lower bounds is large, and it can be seen from Figure 3 that a few such cases were found. Furthermore it is possible in fact to construct matrices for which the gap between mixed μ and the (theoretical) upper bound from theorem 2 is arbitrarily large. In light of this it was decided to investigate the use of Branch and Bound schemes, which take the existing bounds and attempt to improve upon them, and this is discussed in the following section.

7 Improving the Bounds - Branch and Bound Schemes

Whilst the basic concept of a Branch and Bound scheme is very simple, there are several important issues which need to be addressed before such a scheme can be implemented efficiently. In this section we present

some preliminary results from a study of these issues, in the context of mixed μ problems. This will be covered in greater depth in [9].

7.1 Problem Specification

The problem focussed on here is the mixed μ robust stability problem, i.e. given a matrix $M \in \mathbb{C}^{n \times n}$, and an appropriate (mixed) block structure \mathcal{K} , is $\mu_{\mathcal{K}}(M) \leq 1$ (i.e. is the system robustly stable to a given perturbation set)? Since we can only compute bounds for μ in general, we cannot answer this question exactly. Denote the upper bound for μ by μ_{UB} , and the lower bound by μ_{LB} . Then we define upper and lower tolerances, μ_U and μ_L (with $\mu_U > 1$ and $\mu_L < 1$), and we pose the alternative questions; is $\mu_{UB} \leq \mu_U$ and/or is $\mu_{LB} \geq \mu_L$? We will deem our computation scheme to have converged if it has answered either of these two questions.

The motivation for this problem is that an answer to either of these questions effectively answers our original robust stability question within a prespecified tolerance. Establishing that $\mu_{UB} \leq \mu_U$ means that the system is robustly stable to slightly smaller perturbations, whereas establishing that $\mu_{LB} \geq \mu_L$ means that the system is not robustly stable to slightly larger perturbations. Also it is clear that if we can answer these questions with any desired tolerances, then by iterating on this scheme we can compute upper and lower bounds for $\mu_{\mathcal{K}}(M)$ with as small a gap as desired (simply scale the matrix M appropriately and use a bisection type algorithm).

7.2 Issues to be Addressed

The basic idea behind Branch and Bound schemes, in the context of the mixed μ problem, is that one has some algorithm for computing upper and lower bounds for mixed μ , but the bounds may be far apart. In order to refine the bounds one may 'chop' (along one of the real parameters) the subspace of real parameters into two subdomains and then evaluate the bounds on each subdomain (branch). This process is then repeated as often as necessary to refine the bounds as accurately as desired. Note that one has potentially exponential growth in the number of subdomains, but one may eliminate some of the subdomains as the scheme progresses, via standard tree-pruning techniques.

It is of interest to examine the growth rate of computational cost with problem size for such a scheme, and to consider whether such a scheme can be implemented so as to be polynomial time, in worst or typical case. Issues which arise in this regard are the computational cost versus accuracy of the bounds themselves, for any given sub-problem, and also the amount of computational cost one is prepared to pay in order to evaluate a good direction to chop the remaining subspace. In order to address these issues several different Branch and Bound schemes were implemented, and the results obtained with these schemes are discussed in the following subsections.

7.3 Performance of the Various Schemes

The majority of the tests we will describe here were performed using two different Branch and Bound schemes which will be referred to as Scheme A and Scheme B. They were chosen so as to represent two extremes from among the possible choices of schemes. The test matrices described in section 6.3 were used (generated from random 'A,B,C,D' matrices) together with block structures consisting of 2^p real (unrepeated) scalar uncertainties and one $p \times p$ full complex block, for various choices of p (with the aforementioned quantities rounded to the nearest integer). The matrices were randomly scaled with the restriction that μ_{UB} and μ_{LB} , computed using the algorithm from section 6, satisfied $\mu_{UB} > \mu_U$ and $\mu_{LB} < \mu_L$ whenever possible. This is to make the problem as difficult as possible, since otherwise we always obtain an answer to one of our questions immediately.

In Scheme A the upper and lower bounds were computed using the algorithm described in section 6, and as such represent the best bounds we had available. These bounds involve a significant computational cost, though the growth rate appears reasonable (see Figure 1). In addition the choice of which direction to chop was made based on evaluating the bounds in each direction for the chosen subdomain. This increases the cost of computing one branch by a factor m , (the

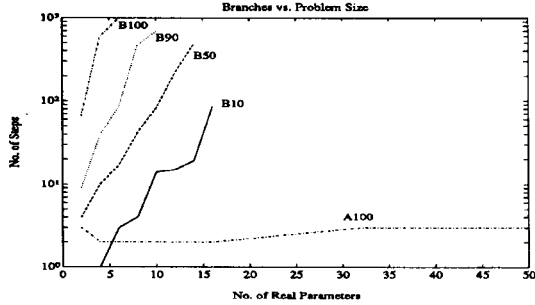


Figure 5: Growth rate of Branch and Bound computation steps versus number of real parameters, for Scheme A and Scheme B.

number of real parameters) which, though significant, is still only a linear increase in complexity. The choice of which subdomain to chop was made based on which had the current highest upper bound, for both Scheme A and Scheme B.

In Scheme B the upper and lower bounds were computed using straightforward norm inequalities on the linear fractional transformation. They involved just norm computations, with no optimization or scaling matrices, and as such represent cheap but crude bounds. The choice of which direction to chop was made by simply selecting randomly from among the longest sides. This is a very cheap 'chopping criterion' and simply ensures that the condition number of the subdomains remains reasonable [5].

A series of tests was performed using these two schemes. The tolerances were chosen as $\mu_L = 0.9$ and $\mu_U = \frac{1}{0.9}$, and the results are shown in Figure 5. For each choice of problem size the two schemes were run on the same 100 test matrices, and the plots indicate the number of 'steps' required by each scheme as a function of problem size. The initial bounds were counted as the first step, and each chop was counted as an additional step. The labeling is best explained by example. The label "B50" means that this is the worst problem encountered by Scheme B, from among the easiest 50% of the problems (for Scheme B). Note that each scheme was allowed a maximum of 1000 steps, to avoid needlessly tying up the computers for days. The curves for scheme B appear to terminate prematurely, but this is because the next points were larger than 1000, and we didn't take data beyond 1000.

It can be readily seen from Figure 5 that the growth rate for Scheme B is exponential on the problem data for any of the levels "B100"- "B10" (note that the results are plotted on a log-linear scale). Scheme A however appears to have a quite reasonable growth rate on these problems, even for the "A100" level. Note also that Scheme B failed to converge (in 1000 steps) on many of the problems, whereas Scheme A converged fairly rapidly on all the problems. These results clearly indicate that it was well worth the extra computational cost of using more sophisticated methods for the bounds and the chopping criterion in our Branch and Bound scheme.

In addition to these results, some studies were performed with other Branch and Bound schemes where the complexity of the bounds and/or the chopping criterion was varied independently. Preliminary results indicate that the contribution of each of these elements to the overall performance is significant. Also the effect of varying the accuracy required for the final answers was examined, and preliminary results indicate that it is probably even more important to use the more sophisticated versions of the bounds and chopping criterion when the required accuracy is increased. These issues will be treated in greater depth in [9].

7.4 A Practical Branch and Bound Scheme

The results from the previous subsection suggest that for a practical Branch and Bound scheme the methods for computing the bounds and the chopping criterion are absolutely critical to the performance on

even medium sized problems. One is prepared to spend a high computational cost on both of these, provided it is still polynomial time, since one is potentially avoiding exponential time growth in the behavior of the Branch and Bound scheme (note that if any branch yields no improvement in the bounds then the subsequent computation can be doubled, since the same computation may have to be performed for each branch). We believe that it is possible to develop a practical Branch and Bound scheme, whose typical behavior on problems of engineering interest has polynomial time growth rate with the problem size. In fact we believe that Scheme A, outlined in the previous section, will form the basis of such a scheme, and work is under way to develop this further. Note that this represents a somewhat different philosophy of Branch and Bound than the one suggested in [5], for example. Here we are performing a relatively small number of quite expensive branches, as opposed to a large number of cheap ones. It should be noted that the approach in [5] might actually work better on small (< 10 parameters) problems.

Finally we note that if one can construct a matrix where the bounds are not within tolerance at the first step (which we can for even our best bounds), then one can build from this examples which will require exponential time growth rate for the Branch and Bound scheme. Essentially what this result says is that if one has to Branch and Bound at all, then it will be exponential time in the worst case. This is not surprising, as our algorithm must not be worst-case polynomial time or it would show that $P=NP$. Of course this does not mean that one cannot have a scheme whose typical behavior is polynomial time, as discussed earlier.

8 Acknowledgements

We would like to acknowledge the help of several of our colleagues. Steve Boyd, Jim Demmel, Andy Packard, Pradeep Pandey, Jie Chen, Michael Fan, and Carl Nett provided us with preprints of their papers and/or copies of software, as well as valuable discussions about the issues in this paper. Bobby Bodenheimer assisted in the numerical experiments. This work was supported by ONR, NSF, NASA, and Rockwell International.

References

- [1] J. Demmel, "The componentwise distance to the nearest singular matrix," to appear in *SIAM Journal on Matrix Analysis and Applications*.
- [2] A. K. Packard and P. Pandey, "Continuity properties of the real/complex structured singular value," submitted to *IEEE Transactions on Automatic Control*.
- [3] P. M. Young and J. C. Doyle, "Computation of μ with real and complex uncertainties," in *Proceedings of the 29th Conference on Decision and Control*, pp. 1230-1235, IEEE, 1990.
- [4] M. K. H. Fan, A. L. Tits, and J. C. Doyle, "Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics," *IEEE Transactions on Automatic Control*, vol. AC-36, pp. 25-38, 1991.
- [5] V. Balakrishnan, S. Boyd, and S. Balemi, "Branch and bound algorithm for computing the minimum stability degree of parameter-dependent linear systems," tech. rep., Information Systems Laboratory, Department of Electrical Engineering, Stanford University, 1991.
- [6] A. Sideris and R. S. Sánchez Peña, "Fast computation of the multivariable stability margin for real interrelated uncertain parameters," *IEEE Transactions on Automatic Control*, vol. 34, pp. 1272-1276, Dec. 1989.
- [7] P. M. Young and J. C. Doyle, "Properties of the mixed μ problem and its bounds," Manuscript in Preparation.
- [8] P. M. Young, M. P. Newlin, and J. C. Doyle, "Practical computation of the mixed μ problem," Manuscript in Preparation.
- [9] M. P. Newlin, P. M. Young, and J. C. Doyle, "Improving the bounds for mixed μ problems via branch and bound techniques," Manuscript in Preparation.
- [10] M. K. H. Fan, A. L. Tits, and J. C. Doyle, "Robustness in the presence of joint parametric uncertainty and unmodeled dynamics," in *Proceedings of the American Control Conference*, pp. 1195-1200, 1988.
- [11] J. Doyle, "Analysis of feedback systems with structured uncertainty," *IEEE Proceedings, Part D*, vol. 129, pp. 242-250, Nov. 1982.
- [12] B. Bamish, P. Khargonekar, Z. Shi, and R. Tempo, "Robustness margin need not be a continuous function of the problem data," *Systems & Control Letters*, vol. 15, pp. 91-98, 1989.
- [13] J. Rohn and S. Poljak, "Radius of nonsingularity," to appear in *Mathematics of Control, Signals and Systems*.
- [14] A. Packard and J. C. Doyle, "Structured singular value with repeated scalar blocks," in *Proceedings of the American Control Conference*, pp. 1213-1218, 1988.
- [15] J. Chen, M. K. H. Fan, and C. N. Nett, "The structured singular value and stability of uncertain polynomials: A missing link," to appear in *ASME annual winter meeting*, Atlanta GA, December 1991.
- [16] A. C. Bartlett, C. V. Hollot, and H. Lin, "Root locations of an entire polytope of polynomials: It suffices to check the edges," in *Mathematics of Control, Signals and Systems*, Springer Verlag, 1988.